

# Responsive Images in Container Queries

Attendees: Brain Kardell, Simon Pieters, Yoav Weiss, Una Kravets

Issue: <https://github.com/w3c/csswg-drafts/issues/5889>

## Concerns:

- Yoav: The only path I see to solve this problem is to add some kind of nested sizes attribute... Like, a sizes attribute for the container, but where the size is about `_that_`. As far as we can tell from content in the wild, people are getting the existing stuff wrong. Maybe if we take that job away from people and gave it to robots...or really good tooling. We have a similar problem is lazy loading where ... well, it feels like a related problem, people want to talk about "the initial viewport" - but there isn't a single one, there are many.
- Simon: Layout needs to be known for the CQ, which would delay the loading of images
  - If you know containers size, couldn't that be calculated based on the viewport?
  - Maybe – considering stable container size per viewport size mapping (Yoav)
- Una: With responsive images now, the image changes as the page resize in `img` attribute, that's what would happen with CW
  - Media attribute this would be a must
  - Picture attribute doesn't do this bc there's no defined callback for when attributes change, and you might not always want to load a new image when the viewport changes (you wouldn't want this to happen for background tabs) (Simon)
- Yoav: There is a right answer, it's just not a markup-based answer
  - Server-side logic that calc all the viewports possible, which images are in the initial viewport for the layout, and then gives you the right answer at server time
  - Impossible to bake this logic into HTML
  - Can do some sizes syntax in HTML
  - Need layout to happen in order for lazyloading and CQ
- Brian: Consider users on low-bandwidth metered connections
  - There are good answers, but there might not be a "right" answer
  - Should you load a new image as things scale? Maybe
- Una: People don't resize screens *that* often to the point that not sending a new image after initial load would necessarily degrade experience
- Yoav: This isn't the main problem, the issue is that these images will be by definition slow
  - Cause adoption issues for container queries, or make the web slower
- Brian: Users open drawers on websites, elements that slide into the layout would be the same problem
- Simon: New images loaded in background (hotswapped)
  - Tricky problem here is figuring out what to do with images in CQ
    - Do we accept these images to load slowly? Nested sizes syntax with container?
- Una: [proposed a container attribute in sizes](#): `sizes="container(max-width: 600px) 480px, 800px"` in the issue

- Simon: requires layout to evaluate this CQ
- Yoav: Another idea: an attribute on the container that's relative to viewport
  - Simon: Can then evaluate CQ before layout is known
  - Yoav: [\[proposed example in jsbin of sizes=\(max-width: 900px\)... with an image inside of it that has container="some syntax"\]](#)
    - Would be for both viewport and container queries
    - People get srcset wrong all the time, we need to give people reasonable tools to generate that
- Brian: What if element is a grid cell
  - No 1:1 mapping from grid to container size
  - This issue exists today with responsive images
  - Grid cell doesn't have a 1:1 mapping to an element, the size can vary
    - Brian: think of it in a table – the app provides the TR, and you give it TD's, they wind up displayed in rows. The available width for whole row but what you're trying to design is a card
    - Size of cell depends on how many cells there are
- Yoav: we need build tools to figure out the perfect sizes value for you, then bake that information back into the HTML
  - We need a tool to use this, or users won't be able to get the size right manually
  - Simon: If you have such a tool, why not just calculate the image size attributes directly
  - Brian: can we reason through one of the use cases and see if Yoav's example even makes sense? One of the features of CQ is that you can reason about a thing without knowing where it will live in the page (without that page)
    - Who would use this tool and how?
    - Can do responsive design today if you start with viewport, why recreate that vs focus on container queries
      - Simon: it still lets you use container queries and not think about the viewport while designing. The consumer of the component may have to use sizes attribute on the container so the browser can load the appropriate image before it knows the layout
        - Goal: load CSS and image in parallel
- Yoav: we tell people they need to run this markup through optimization tool that takes the viewport into account (initial size of container relative to viewport) – not a task for humans, essentially recognizing current status quo (i.e. sizes isn't usually very accurate)
  - Suggesting: we will have some indicator on the image that it's a child of a CQ, or on the container that it's a container, and exclude images within it from preload scanning unless there's an exclusive sizes attribute. And have sizes attributes that are generated somehow
- Simon: Is it better, if you have to wait until layout to load the wrong image or to just wait.  
Yoav: It is a gamble.
- Brian: I think like I said there are good answers, not perfect ones
- Una: I don't think people expect it to be perfect, I think that mostly people want to avoid worst cases - sending a giant image to a tiny device or something.

- Yoav: It might be interesting to do a study - there have been some... cloundinary did one to gather up stats about what happens in the wild today vs final layout dimensions. It could be a good way to get insight into how well we're even doing today. The use cases for the picture media query on the container queries - you need to be right. With srcset, the worst case scenario is you might get the wrong one (first?).
- Una: Another example use case to consider: Sometimes you want a different aspect ratio or even differnt media for different screen sizes - you want different art direction
- Yoav: That would require a perfect sizes attribute or some tooling or people will get this wrong. I think in a way what we were suggesting was that maybe container queries could be built or compiled and I'm not really sure that is what anyone wants.
- Una: I wonder if there is a world in which you can use responsive images in container queries in which this will happen after layout. They can use placeholders for those images, for example... I don't know if that is the worst thing.
- Yoav: It will be slow. But we see now with lazy loading people are sprinkling it everywhere that people are making their pages effectively and measuably slower. Essentially, if you make people between what is easy and what is good for users - they will go with easy every single time. We can do some education, but ultimately I think this will not help us. I think we need to figure out a way in which we can give people actionable advice that is not super hard that gets them the right thing.
  - CSS images in general are loaded late
  - If we say that we need the layout before downloading the image, we have a similar situation w/what we have for background images today
  - Can preload critical bg images, or the best practice is to make them decorative
- Una: Shopping example (products in a grid) grid → hero → cart component style
  - Without responsive images, really degrades experience/usefulness of container queries
- AI:
  - Schedule another 30 mins in a week to continue discussion